Open NFC - Android ICS 4.0.1 - Quick Porting Guide

Document Type:	Manual
Reference:	MAN_NFC_1112-289 Version 0.1 (12361)
Release Date:	Dec. 13, 2011
File Name:	MAN_NFC_1112-289 Open NFC - Android ICS 4.0.1 - Quick Porting Guide v0.1.pdf
Security Level:	General Business Use

Disclaimer

History

Version	Date	Comments
0.1	Dec. 13, 2011	First Release

Summary of Contents

Reference Documents

[1] FSP_NFC_0709-014 **Open NFC Core Edition - Porting Guide**

[2] FSP_NFC_0707-003 **Open NFC – API Specification**

[3] MAN_NFC_0711-028 **Open NFC Linux Edition – Porting Guide**

1 Introduction

This document is the quick version of Android porting guide for the NFC software stack "Open NFC v4.3.3 ICS".

This document describes how the porting described in document reference [1] has been done for the Android Gingerbread platform release 4.0.1_r1 and how it must be adapted for a real hardware platform.
The main components of the android porting are:

An Android service, named OpenNFCService, which controls the NFC controller and deals with the multi-application management. This service is bundled with a sample application, named OpenNfcSettings (Inside packages/apps/OpenNFC/frameworks/OpenNFCService/src : org.opennfc.tools.opennfcsettings.OpenNfcSettings), used for the management (start/stop, …) of the service

A driver, named open_nfc_driver.ko, used for the low level communication with the NFC controller.

2 Building Instructions

How to compile the Android sources with Open NFC

The following patches are necessary :

- build.patch
- frameworks_base.patch
- system_core.patch

You will find it in Open NFC package in the sub-directory `patches`, see below Open NFC's tree.

2.1 Create bin/ directory and include it in path

```
$ mkdir ~/bin
$ PATH=~/bin:$PATH
```

2.2 Download the Repo script and ensure it is executable

```
$ curl https://dl-ssl.google.com/dl/googlesource/git-repo/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

2.3 Make a New working directory

```
mkdir android-4.0.1_r1 && cd android-4.0.1_r1
```

2.4 Run repo init to bring down the latest version of Repo

```
$ repo init -u https://android.googlesource.com/platform/manifest -b android-4.0.1_r1
```

2.5 Run repo sync to pull down files

```
$ repo sync
```

2.6 Getting last Open NFC on SourceForge

Create directory packages/apps/OpenNFC

```
$ cd packages/apps
$ mkdir OpenNFC && cd OpenNFC
$ git clone git://open-nfc.git.sourceforge.net/gitroot/open-nfc/open_nfc_4.3.3_ics.git .
```

2.7 Apply build.patch

```
android-4.0.1_r1/build$ cp <your_path_directory_patches>/build.patch  .
android-4.0.1_r1/build$ git apply --check build.patch
android-4.0.1_r1/build$ git apply --stat build.patch
core/base rules.mk |    2 +-
 core/main.mk      |    4 +++-
 2 files changed, 4 insertions(+), 2 deletions(-)

android-4.0.1 r1/build$ git status
# Not currently on any branch.
# Untracked files:
```

```
#   (use "git add <file>..." to include in what will be committed)
#
#       build.patch
nothing added to commit but untracked files present (use "git add" to track)

android-4.0.1_r1/build$ git apply --whitespace=nowarn build.patch
android-4.0.1 r1/build$ git status
# Not currently on any branch.
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#             modified:   core/base rules.mk
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#             build.patch
no changes added to commit (use "git add" and/or "git commit -a")
```

2.8 Apply system_core.patch

```
android-4.0.1 r1/system/core$ cp <your path directory patches>/system core.patch .
android-4.0.1 r1/system/core$ git apply --check system core.patch
android-4.0.1 r1/system/core$ git apply --stat system core.patch
rootdir/init.rc    |    3 +++
rootdir/ueventd.rc |    3 +++
2 files changed, 6 insertions(+), 0 deletions(-)
android-4.0.1 r1/system/core$ git apply --whitespace=nowarn system core.patch
android-4.0.1 r1/system/core$ git status
# Not currently on any branch.
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#             modified:   rootdir/init.rc
#             modified:   rootdir/ueventd.rc
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#             system core.patch
no changes added to commit (use "git add" and/or "git commit -a")
```

2.9 Apply the patch frameworks_base.patch

```
android-4.0.1 r1/frameworks/base$ cp <your path directory patches>/frameworks base.patch .
fheiser@android:/mnt/workspace/android-4.0.1 r1/frameworks/base$ git apply --check
frameworks base.patch
fheiser@android:/mnt/workspace/android-4.0.1_r1/frameworks/base$ git apply --stat
frameworks_base.patch
data/etc/platform.xml |    5 +++++
1 files changed, 5 insertions(+), 0 deletions(-)
android-4.0.1_r1/frameworks/base$ git apply --whitespace=nowarn frameworks_base.patch
```

2.10 Check in tree

In <ANDROID_ROOT>/packages/apps you must have the following hierarchy :
(Hints: In <ANDROID_ROOT>/packages/apps you can use tree –d OpenNFC)

```
OpenNFC/
├── frameworks
│   ├── libs
```

```
│   │   └── open_nfc
│   │       ├── interfaces
│   │       ├── porting
│   │       │   ├── client
│   │       │   │   └── jni
│   │       │   │       └── headers
│   │       │   ├── common
│   │       │   └── server
│   │       │       └── jni
│   │       └── sources
│   ├── OpenNFCLibrary
│   │   ├── docs
│   │   │   └── html
│   │   │       ├── Explanations_files
│   │   │       └── reference
│   │   │           └── org
│   │   │               └── opennfc
│   │   │                   ├── cardemulation
│   │   │                   │   └── doc-files
│   │   │                   ├── cardlistener
│   │   │                   │   └── doc-files
│   │   │                   └── nfctag
│   │   │                       └── doc-files
│   │   └── java
│   │       └── org
│   │           ├── opennfc
│   │           │   ├── android
│   │           │   │   └── tools
│   │           │   ├── cardemulation
│   │           │   │   └── doc-files
│   │           │   ├── cardlistener
│   │           │   │   └── doc-files
│   │           │   ├── hardware
│   │           │   ├── nfctag
│   │           │   │   └── doc-files
│   │           │   ├── p2p
│   │           │   ├── p2pHo
│   │           │   └── se
│   │           └── simalliance
│   │               └── openmobileapi
│   └── OpenNFCService
│       ├── assets
│       ├── res
│       │   ├── drawable-hdpi
│       │   ├── drawable-mdpi
│       │   ├── layout
│       │   ├── raw
│       │   ├── values
│       │   └── xml
│       └── src
│           └── org
│               └── opennfc
│                   ├── service
│                   │   ├── communication
│                   │   │   ├── npp
│                   │   │   └── snep
│                   │   ├── p2pEvent
│                   │   └── P2PHandover
│                   ├── tests
│                   │   ├── generic
│                   │   ├── TestLibJava
│                   │   │   ├── NdefMessage
│                   │   │   ├── NdefRecord
│                   │   │   ├── NfcTagConnection
│                   │   │   └── utils
│                   │   │       └── tag
│                   │   └── unit
│                   └── tools
│                       ├── opennfcsettings
│                       └── securityStack
├── hardware
│   └── libhardware
│       ├── include
│       │   └── hardware
│       └── modules
│           └── nfcc
```

```
│                ├── nfcc
│                ├── nfc_hal_microread
│                │   ├── driver
│                │   ├── interfaces
│                │   ├── porting
│                │   └── sources
│                │
│                └── nfc_hal_simulator
│                    ├── interfaces
│                    ├── porting
│                    └── sources
└── patches
```

2.11 Kernel Compilation

Firstly, you should download the source the source of the linux kernel:

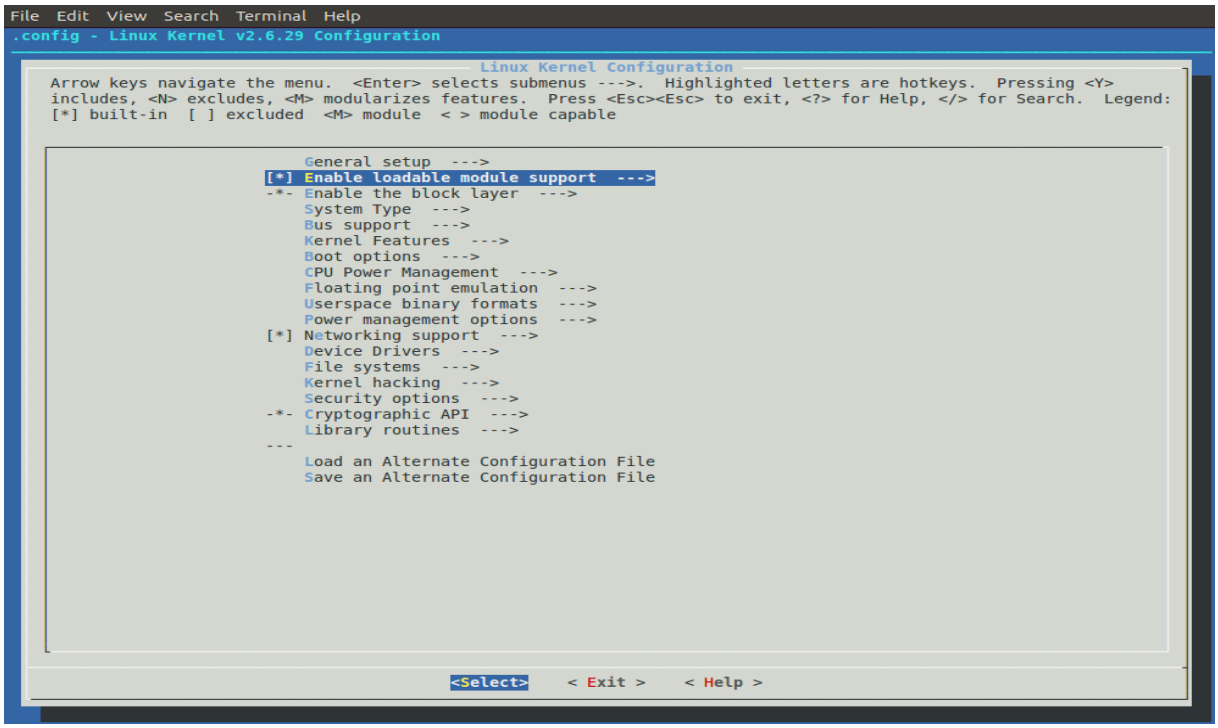
```
mkdir kernel
cd kernel
git clone https://android.googlesource.com/kernel/goldfish.git .
git checkout origin/android-goldfish-2.6.29
```

The default android emulator Linux kernel configuration can be generated using the following commands:

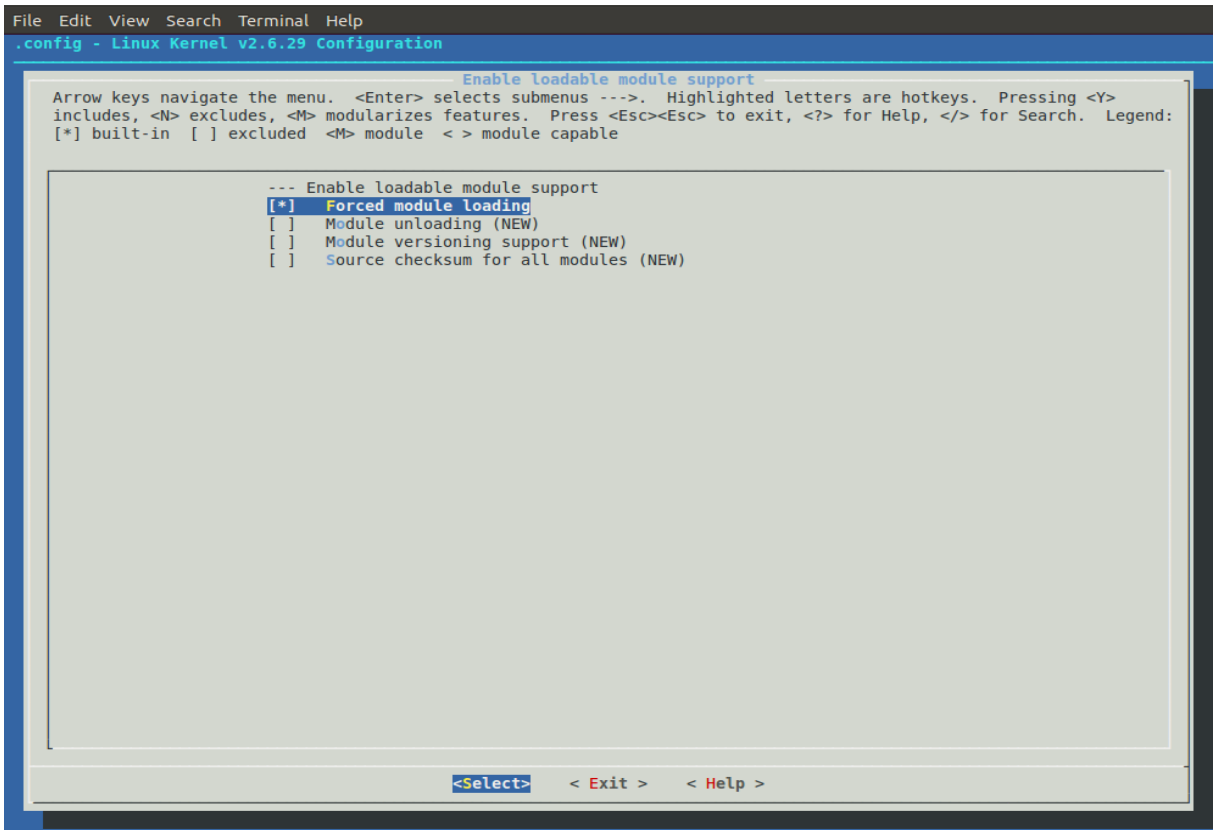
```
export CROSS_COMPILE=arm-eabi- ARCH=arm
cd (ANDROID_ROOT_DIR)/kernel
make goldfish_armv7_defconfig
```

Then, enable the loadable support module.

```
make menuconfig
```



Choose “Enable loadable module support” and press <Space>



Enable “Forced module loading”

Then, compile the new kernel

```
$ make -j`cat /proc/cpuinfo | grep -E '^processor' | wc -l`
```

This kernel must be passed to the emulator command using the –kernel option.

```
emulator -kernel kernel/arch/arm/boot/zImage
```

The open_nfc_driver kernel sources can be found in the directory:

```
$ export ANDROID_BUILD_ROOT=$(ANDROID_ROOT_DIR)
$ cd
(ANDROID_ROOT_DIR)/packages/apps/OpenNFC/hardware/libhardware/modules/nfcc/
nfc_hal_microread/driver
```

Assuming the kernel sources have been downloaded into the (ANDROID_ROOT_DIR)/kernel directory, the kernel module compilation can be done by issuing a make command in the driver directory.

```
$ make -j`cat /proc/cpuinfo | grep -E '^processor' | wc -l`
```

2.12 Final Compilation

Once Open NFC Android successfully installed in the standard AOSP sources, lunch command in the AOSP tree:

```
$ . ./build/envsetup.sh

$ lunch

You're building on Linux
```

```
Lunch menu... pick a combo:
      1. full-eng
     2. full_x86-eng
     3. vbox_x86-eng
     4. full_maguro-userdebug
     5. full_tuna-userdebug
     6. full_panda-eng
Which would you like? [full-eng] 1

$ make –j2 ARCH=arm CROSS_COMPILE=arm-eabi-
```

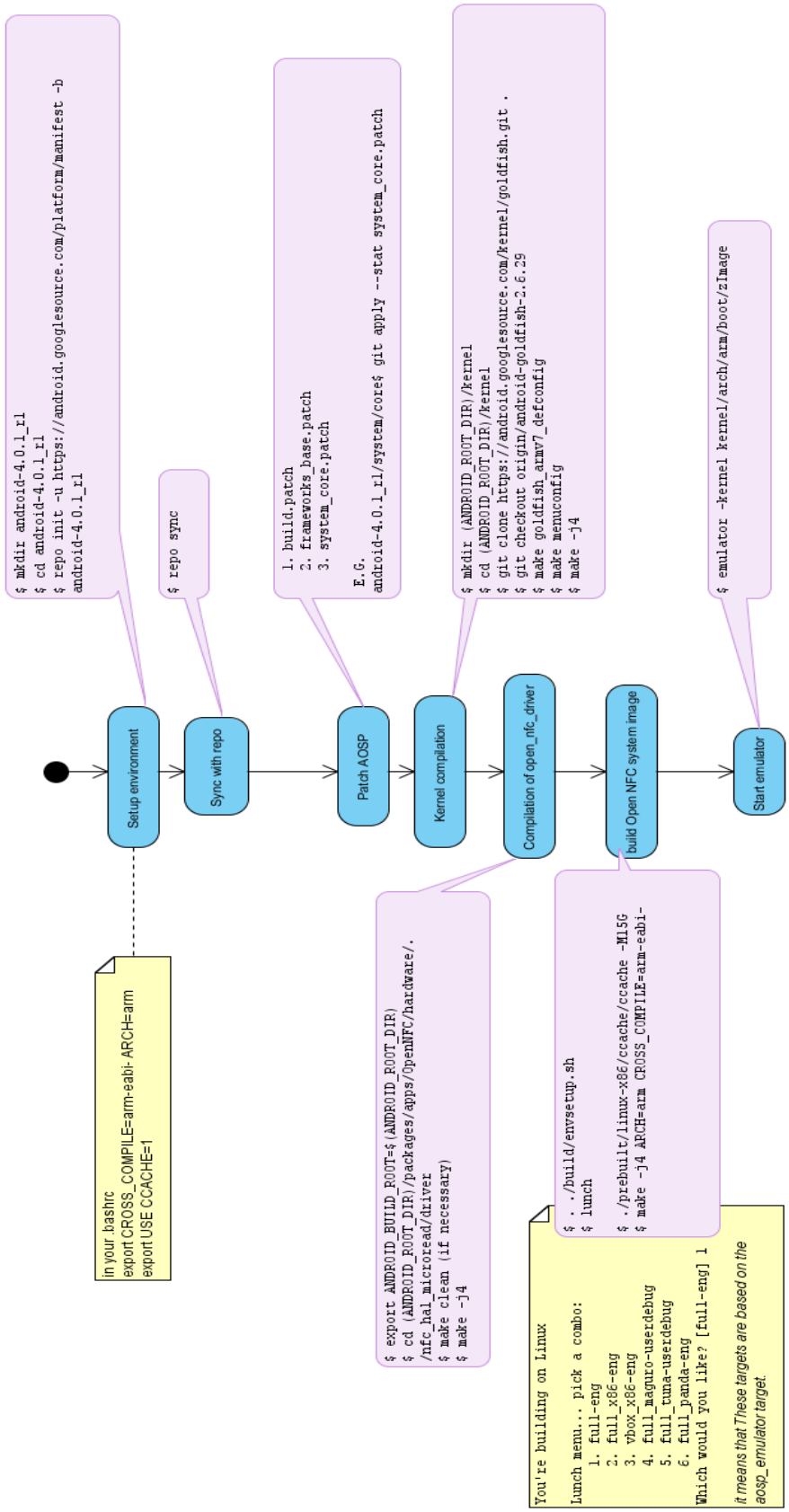
Tips: These targets are based on the aosp_emulator target.

Once the correct target selected, a simple make command on the android top dir will build the entire project.

Recommendation: enable CCACHE and export the following variable USE_CCACHE and enhance the ccache size following your available space disk.

```
$ export USE_CCACHE=1
$ ./prebuilt/linux-x86/ccache/ccache –M15G
```

3 Annex 1



4 License

The source files of the driver sample of the "Open NFC for Linux" are distributed using the "Apache v2.0" license. They can be freely adapted or modified and used to create a dynamic driver module or to be statically linked with the kernel without any license issue.

The remaining of the Open NFC core source code and the source code of the "Open NFC for Android" are distributed under Apache v2.0 license.